

HOW IT WORKS

TCP/IP

The internet and, increasingly, our homes rely on the Internet Protocol to let devices talk to each other. David Ludlow explains the network technology behind the web

The internet owes its fast and flexible communication abilities to the network technology behind it: transmission control protocol (TCP) and internet protocol (IP). IP is responsible for sending data between computers, while TCP handles stability, connections and error control. It's a great combination, and a successful one that enables mobile phones, PCs, Apple computers and even fridges to communicate seamlessly with each other and share information with ease.

LOCATION, LOCATION, LOCATION

IP is the starting point for communication. Every computer connected to the internet has an IP address that gives its location. An IP address is generally written down as four sets of three-digit numbers separated by full stops, such as 192.168.123.129. This representation is designed to be easier for people to read; computers actually use IP addresses in binary form.

Each of the four numbers in the human-friendly decimal version equates to an eight-bit binary number. As binary is base two, each quarter of an IP address has a maximum value of 255 (2^8 gives us 256 numbers from zero to 255). However, addresses that end with the three-digit number 255 are reserved for broadcast traffic to the entire network. Computers don't recognise the full-stops, so they see an IP address as a single 32-bit binary number. Our example IP address is actually:

```
192      168      123      129
11000000 10101000 01111011 10000001
```

which gives us a total of 4,294,967,296 unique addresses (2^{32}). This may sound like a lot, but many addresses are reserved for particular uses. For example, all addresses that start 127.x.x.x are for internal testing. Data sent to these addresses is not sent out, but is processed locally.

While reserving addresses reduces the pool of available addresses, techniques such as network address translation (NAT) can help. Broadband routers commonly use NAT, which enables all your PCs to share a single unique IP address on the internet. Behind the router, however, each computer has its own private IP address to differentiate it from other PCs on your network. Any request that a computer makes over the internet is translated by the router so that, as far as the rest of the internet is concerned, it originates from your unique internet IP address.

NETWORK AND HOST

IP addresses are actually a little bit more complex than this, though. They are in fact made of two elements joined together: the network address and the host address. You can think of the network address as the town you live in,

while the host address is your street address. When information is sent across the internet, the host address is initially ignored. Internet routers that tell the traffic which direction to take simply look at the network address. Once the data turns up at the correct network, the network address is ignored and the data is sent to the correct host.

Network addresses are divided into classes of network. The class determines how many of these networks there are and how many host addresses are available for computers. Three classes of network are commonly used, which are referred to as A, B and C. In terms of the number of addresses available, the three classes of network work out like this:

Class	Number of networks	Number of hosts
A	126	16,777,216
B	16,384	65,356
C	2,097,152	256

In binary, class A addresses start with 0, class B start with 10, and class C start with 110. Our example IP address is a class C address.

Even a class C network can be too big for some people, so networks can be split into smaller segments using a process called subnetting. This works by using a given number of bits of the binary host address for subnet information. So, if we wanted to give two bits of our class C address to the subnet, we could split our network into four virtual networks (subnet addresses of 00, 10, 01 and 11). However, using two bits of the host address for the subnet means that we now have only six bits for hosts, so each of the four subnets can have only 62 hosts in it — 2^6 equals 64 addresses, but the first address is called the network address and last address is for broadcasting to the entire network.

Assuming we use two bits for a subnet, our example IP address can now be viewed like this:

Network address	Subnet	Host
110000001010100001111011	10	000001

The subnet mask allows us to determine which computers can communicate with each other. Only computers on the same subnet can communicate directly. If you click on Run from your PC's Start menu, start a Command prompt, type ipconfig and hit Enter, you'll see your PC's subnet mask. This will probably be a number like 255.255.255.0.

So how is this number calculated? If we look at it in binary form, we can see that all the digits of the network address have been converted to 1s, as have the subnet's digits, while all the host address's digits have been converted to 0s. With the example above we get a subnet mask of:

```
11111111 11111111 11111111 11000000
```

Converting this back to decimal, we've got a subnet mask of 255.255.255.192.

To find out if two computers are on the same subnet, we perform a Boolean AND operation on each address with the subnet mask. Boolean AND works in the following way: 0 AND 0 = 0; 1 AND 0 = 0; 0 AND 1 = 0; and 1 AND 1 = 1.

If we were to take two IP addresses, 192.168.123.129 and 192.168.123.130, and perform a Boolean AND with a subnet mask of 255.255.255.192 with each of them in binary, we'd get:

```
192.168.123.129:
11000000 10101000 01111011 10000001
AND
11111111 11111111 11111111 11000000
Equals
11000000 10101000 01111011 10000000
```

```
192.168.123.130:
11000000 10101000 01111011 10000001
AND
11111111 11111111 11111111 11000000
Equals
11000000 10101000 01111011 10000000
```

As both calculations give us the same answer (the network address, 192.168.123.128), we know that both addresses are on the same subnet and can talk to each other. If the second IP address had been 192.168.123.10, a Boolean AND with 255.255.255.192 would have given us 192.168.123.0, putting it on a different subnet.

ROUTING AROUND

When sending data to a computer on the same network, or even one on the other side of the world, IP comes into play again. A computer splits the data it wants to send into small fragments called packets. This way, if corruption occurs, only a small amount of data is lost and has to be resent.

The diagram of the IP packet (see opposite) shows all the information that is transmitted. A packet is 32 bits wide. To begin with, the source and destination addresses are the most important part. The source address says who's sending the packet, so that the receiver can reply; the destination address says where the packet is going.

If the source and destination addresses are on the same network, the transition is handled internally. If not, the packet is sent to a router. This examines the destination address and looks up a route in its routing table, which tells it how to reach any network on the internet. As the internet is really a series of interconnected networks, there could be hundreds, if not thousands, of ways for a packet to travel to its destination. To help, the routing table also ranks

each link to which it's connected, based on link speed and delay, among other criteria. This information is updated frequently so that the packet can always take the best route.

PACKET IN

As the packet diagram (below right) shows, lots of different bits of information are added to the start of the data in what's known as the header. First, the version field contains a four-bit number that says which version of IP is being used. This is IPv4 at the moment, although IPv6 is gaining some support. The IP header length (IHL) field states how long the header is. These can vary in length because of the options field later.

The type of service (TOS) field is supposed to determine the level of service that a packet gets. In practice, other techniques are used to prioritise the traffic instead. The total length field states the size of the packet in bytes, including the header. As it's a 16-bit number, the maximum length of a packet is 65,535 bytes.

Some links along the route have a maximum packet size. Any packets that are larger than this limit are split up further, which is known as fragmenting. The identification field tells a receiving computer which packet a fragment belongs to. All fragments from the same packet have the same identification value.

If fragmentation is unacceptable, the don't fragment (DF) field can be set. This stops routers splitting up the data, but it can mean that the packet must take a slower route. When data is allowed to fragment, the more fragments (MF) field tells the receiving computer that more fragments are on the way. All fragments except the last are marked with this field.

To help a computer reassemble these fragments, the fragment offset field tells it where in the original packet the fragment belongs. This means that fragments can take different routes and arrive out of order, but still be reassembled. All fragments except the last must be a multiple of eight bytes long.

As there are many routes a packet can take across the internet, it's possible for one to end up in an infinite loop. The time to live (TTL) field stops this happening. Each router that the packet passes through (called a hop) reduces the TTL by one. When it hits zero, the packet is deleted.

The protocol field says which protocol is used inside the packet. TCP is very common (see below), as is UDP.

The header checksum is used to verify the data. If it doesn't tally with the data, the packet is deleted. Finally, the options field lets certain options be specified, such as the route to take.

TRANSMISSION CONTROL

IP has nothing to do with managing a connection and doesn't care if packets are delivered or not. TCP is there to make sure data is sent. A TCP packet is sent in the data field of an IP packet and adds another header (see the diagram, right).

A TCP port is a special number that maps data to a particular process running on a computer. Services use specific ports; for example, FTP traffic uses port 21. The source port field gives the port on which the transmitting computer expects a reply. The destination port field says which port on the receiving computer it's trying to talk to, such as port 80 for web traffic.

Each packet is given a sequence number to ensure that data doesn't get lost. When a

computer receives a packet, it sends an acknowledgement and fills in the acknowledgement number field with the sequence number of the packet it expects to receive next (usually, the last sequence number plus one). If a packet goes missing, this acknowledgement number will be asking for a packet that has already been sent, so the sending computer can go back and resend this information. This ensures that the complete data is transmitted and received.

The header length (HLN) field states how long the header is, as the options field can vary in length. The reserved field isn't currently used.

The next six fields are control bits. URG is used to mark urgent data. ACK is an acknowledgement. PSH tells the receiver to push the data to the application as soon as possible, even cancelling an application's job early. SYN is used to synchronise the sender and receiver. FIN tells the receiver that there's no more data.

SYN and ACK are the most commonly used fields. When a computer makes a connection with a server to get a web page, for example, it sends a SYN request. The server responds with its own SYN and an ACK. The sender responds with a final ACK, and the connection is established and ready for data transfer.

The window field is used for flow control and states how much data the receiver can accept. Reducing this number slows down the connection speed; increasing it speeds up communication. Its primary use is to stop a receiver getting flooded by too much data.

The checksum field is used to verify the data. The receiver performs its own checksum calculation on the data and compares it with the figure in the checksum field; if the result differs, the packet is discarded.

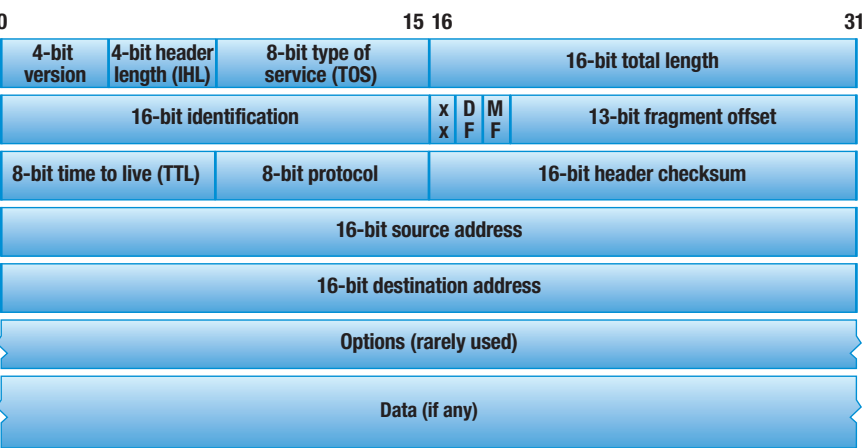
The urgent pointer is used with the URG bit, and tells the receiver where the last byte of the urgent data ends. The options field can vary in length and contains transmission options, such as for congestion and flow control. Finally, the data field carries the actual data, if there is any.

FULL SPEED AHEAD

We won't go into the full details here, but there are times when it's more important to send data quickly rather than reliably, such as for video and phone calls. In these situations, TCP's error checking and flow controls aren't needed, so UDP is used instead. This protocol doesn't control the connection or resend packets, so you get quick transmission, although errors can be introduced. 

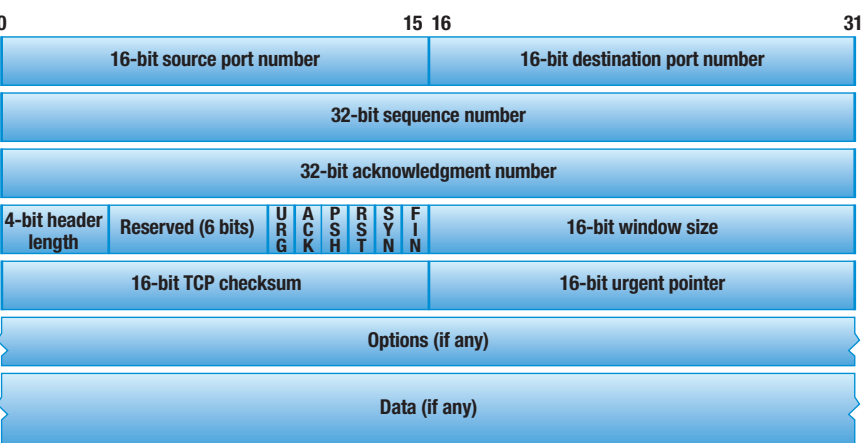
Heads up IP and TCP headers

IP HEADER



▲ As well as the actual data, an IP packet contains a header. This gives information such as the source and destination address, the header's length and whether the packet has been fragmented

TCP HEADER



▲ TCP has its own header within the IP packet's data field. This enables computers to establish a connection, keep track of the packets sent and, if necessary, resend any that get lost on the way

DIAGRAM: MIKE HARDING